

# Projeto

## NETFLOX

Bases de Dados

Departamento de Engenharia Informática



### Objetivos:

- Criação de uma aplicação de bases de dados com uso da linguagem Python

### Requisitos:

- Python 3.8.5
- PostgreSQL 12
- Psycopg2 2.8.6
- PyCharm
- Acesso à ferramenta ONDA em <https://onda.dei.uc.pt>

### Material de Apoio:

- Slides das aulas e referências sugeridas.

### Descrição

A empresa *NETFLOX* pretende um sistema que lhe permita gerir o seu negócio online, que consiste essencialmente em alugar artigos de vários tipos (séries, filmes e documentários) para visualização online e gerir o seu inventário de artigos disponíveis para alugar. Assim, é necessário construir uma aplicação de bases de dados que permita que clientes possam, por exemplo, pesquisar e alugar artigos disponíveis, e que permita que os administradores, que são manualmente adicionados à base de dados, possam desempenhar tarefas típicas de gestão do sistema (e.g., adicionar um novo artigo, ou visualizar estatísticas). Segue-se uma breve descrição das funcionalidades a implementar.

### Cliente:

- Registrar cliente. Requer nome, endereço de email e password (deve ser encriptada de forma a garantir a segurança). A cada registo novo a loja atribui um saldo inicial por omissão (e.g., 20 €) que o cliente usará em alugueres de artigos.
- Login (o nome do cliente deve ser visível nos menus disponíveis após o login) / Logout.
- Listar todos os artigos.
- Ver os detalhes de um artigo.
- Alugar um artigo, mostrando o preço e o tempo que terá o artigo disponível para visualização (por exemplo, o aluguer de uma série pode ter 6 meses, mas o aluguer de um filme poder ter uma semana). Deve existir algum mecanismo que impeça o aluguer, se o saldo não for suficiente.
- Listar todos os artigos que tem neste momento disponíveis para ver, bem como todos os que alugou no passado.
- Mostrar quanto gastou em alugueres passados, fazendo também distinção do valor gasto por cada tipo de artigo.
- Consultar mensagens enviadas pelo administrador, deve ser possível distinguir as lidas das não lidas.

– Pesquisar artigos usando diversos critérios: por tipo, por título, por atores, por realizador, por produtor. Deve ser possível especificar critérios de ordenação dos resultados. Esta funcionalidade deve ser aplicável em dois contextos: i) a todos os artigos no sistema; ii) aos artigos neste momento alugados pelo cliente.

#### **Administrador:**

- Login via email e password / Logout.
- Adicionar um novo artigo.
- Visualizar todos os artigos disponíveis (incluindo preço e condições de aluguer).
- Corrigir o preço de um artigo (deve ser mantido um histórico com as correções, que também é visto quando se visualizam os detalhes de um artigo).
- Remover um artigo (apenas se não foi ainda alugado por nenhum cliente).
- Enviar uma mensagem a todos os clientes.
- Enviar uma mensagem a um cliente específico.
- Aumentar o saldo de um certo cliente (assume-se que os clientes enviam o valor para o administrador que altera manualmente cada saldo).
- Ver estatísticas: total de clientes, totais de artigos, valor total dos artigos alugados neste momento, valor total dos alugueres, total de artigos por tipo. Para além destes valores indicados, apresente mais dois valores à sua escolha, que considere úteis num sistema deste tipo.

#### **Datas importantes**

##### **Entrega I – 23h59, 6 de Novembro de 2020**

Para esta entrega deve produzir um relatório curto que apresente o trabalho efetuado e o plano para a entrega II. Assim, o relatório deve incluir:

- Especificação da interface com o utilizador (deve ser baseada em texto) para todas as funcionalidades
- Imagem do diagrama Entidade Relacionamento
- Imagem do diagrama Físico (versão *simplified* no ONDA)
- Distribuição planeada de tarefas por cada elemento do grupo para a entrega II.
- O tempo gasto com a disciplina \*por semana\* (e por aluno) desde o início do semestre:
  - Em presenças nas aulas T e PL.
  - Extra-aula.

Apresente também numericamente o total de horas de esforço gastas até esta entrega.

- Outros aspetos que o grupo considere relevantes.

##### **Entrega II – 23h59, 11 de Dezembro de 2020**

Nesta entrega deve ter todas as funcionalidades implementadas, incluindo o uso, a título de exemplo, de:

- i) uma transação e
- ii) uma função (ou procedimento ou *trigger*) em PL/SQL.

Deve também apresentar uma versão melhorada do relatório anterior. Deve entregar num zip:

- Todo o código desenvolvido para este projeto, incluindo o ficheiro JSON com o modelo de dados criado no ONDA e o script de criação de tabelas correspondente.
- Relatório curto, que deve incluir:
  - Organização final geral da aplicação (e.g., ficheiros e suas relações).
  - Alguns *screenshots* ilustrativos do aspeto geral da aplicação e das suas partes mais importantes.
  - Diagrama Entidade Relacionamento (indicando que alterações sofreu desde a entrega I, se existirem).
  - Diagrama físico, versão *simplified* no ONDA (indicando que alterações sofreu desde a entrega I, se existirem).
  - Distribuição final de tarefas por cada elemento do grupo.
  - O tempo gasto com a disciplina \*por semana\* (e por aluno) desde o início do semestre:
    - Em presenças nas aulas T e PL.
    - Extra-aula.

Apresente também numericamente o total de horas de esforço gastas até esta entrega.

- Outros aspetos que o grupo considere relevantes.

### Notas finais

– Os projetos são realizados em grupos de **2 alunos**, sendo as duas entregas posteriormente acompanhadas de duas defesas orais. Antes de cada entrega serão abertos *slots* no inforestudante que os grupos devem usar para fazer a sua inscrição.

– Se identificar um erro na aplicação ONDA (ou se tiver sugestões), p.f. envie de imediato um email para **onda@dei.uc.pt**, explicando o problema, para que possa ser corrigido em tempo útil. Agradecemos o seu feedback valioso.